OLLSCOIL NA hÉIREANN, MÁ NUAD

NATIONAL UNIVERSITY OF IRELAND, MAYNOOTH

M.Sc./H.Dip. SOFTWARE ENGINEERING EXAMINATION

SAMPLE 2001/2002

PAPER CS605

MATHEMATICS AND THEORY OF COMPUTER SCIENCE

Mr. T. Naughton

Answer four (4) questions from six (6). Time Allowed: 3 hours.

NOTE: Although this sample paper contains more than 6 questions, in the exam you will be given only 6 questions to choose from.

1. Prove that the positive rationals are countable. Why can we not use diagonalisation to show that the positive rationals are uncountable? [25 marks]

2. Compare and contrast the regular languages and the context-free languages. How are they related to the capabilities of various machine models? Give examples as necessary. [25 marks]

3. (a) Define two languages that are context-free but not regular. [6 marks]

   (b) Construct two nondeterministic pushdown automata to accept these languages [i.e. one for each language from (a)]. [14 marks]

   (c) Let $\Sigma = \{a, b\}$. Is Σ^* finite? Is Σ^* countable? [5 marks]

4. Let A, B, and C be three countably infinite sets. Let $X = A \cup B \cup C$.

   (a) Prove that X is countable. [10 marks]

   (b) Prove that 2^X is uncountable. [10 marks]

   (c) Let $Y = A \times B \times C$. Is Y countable? Explain. [5 marks]

5. (a) What does it mean for a set L to be a language over another set Σ? Define the three smallest languages over a Σ of your choice. [5 marks]

   (b) For all sets Σ and all sets L, is it possible to decide if a given L is a language over a given Σ? Explain. Restrict Σ to being a finite set. What is your answer now? Explain. [15 marks]

   (c) Let $\mathcal{CF}$ be the set of context-free languages. What does it mean for a language to be $\mathcal{CF}$-complete? Illustrate your answer with an example of a $\mathcal{CF}$-complete language. [5 marks]

6. Consider the following deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F) = (\{00, 01, 02, 03, 04\}, \{a, b\}, \delta, 00, \{02, 03\})$ where δ is given by

q	s	q'
00	a	01
00	b	02
01	a	01
01	b	03
02	a	04
02	b	04
03	a	04
03	b	04
04	a	04
04	b	04

(a) Give a regular expression for the language accepted by M. [6 marks]

(b) Using the style used to define M, define a nondeterministic finite automaton that accepts the same language. [7 marks]

(c) Let $\mathcal{REG}$ be the set of regular languages. Let $\mathcal{CF}$ be the set of context-free languages. Is $\mathcal{REG} \subseteq \mathcal{CF}$, or is $\mathcal{CF} \subseteq \mathcal{REG}$, or both? Explain. [6 marks]

(d) Given an arbitrary set X, under what conditions will 2^X be countable? Under what conditions will 2^X be finite? [6 marks]

7. Consider the following nondeterministic pushdown automaton $M = (Q, \Sigma, \Delta, q_0, F) = (\{00, 01\}, \{a, b, c\}, \Delta, 00, \{00, 01\})$ where Δ is given by

q	s	pop	q'	psh
00	a	e	00	a
00	b	e	00	b
00	c	e	01	e
01	a	a	01	e
01	b	b	01	e

(a) Give a context-free grammar that generates the language accepted by M. [6 marks]

(b) Let $\mathcal{CF}$ be the set of context-free languages. Let $\mathcal{R}$ be the set of recursive languages. Is $\mathcal{CF} \subseteq \mathcal{R}$, or is $\mathcal{R} \subseteq \mathcal{CF}$, or both? Explain. [6 marks]

(c) A University has n clubs and societies, the largest of which contains m members (of course, students can be members of multiple clubs and societies). The President of the University wishes to hold a dinner in honour of such student activities. Unfortunately, Pugin Hall can seat comfortably only k guests. The President's problem is as follows: can he construct a guest list of k students such that every club and society has at least one member in attendance? You must prove that this problem is $\mathcal{NP}$-complete. You are given only that the problem SATISFIABILITY is $\mathcal{NP}$-complete. [13 marks]

8. (a) Let Σ be a finite alphabet. State and explain the difference (in terms of computability) between the task of enumerating all of the words over Σ and the task of enumerating all of the languages over Σ. [10 marks]

(b) Construct a nondeterministic two-stack pushdown automaton to accept the language $L = \{w : w \in \{a, b\}^*, w = a^n b^{2n} a^n, n \in \mathbb{N}, n \geq 0\}$. [15 marks]

9. You are given a two-tape Turing machine $T = (Q, \Sigma, I, q_0, F) = (\{00, 01, 02, 03, 09\}, \{0, 1, -\}, I, 00, \{09\})$ that operates on binary strings. The head of the first tape will be positioned at the beginning of the input and the second tape will be blank. I is

q	s	q'	s'	m
00	$(0, -)$	00	$(0, -)$	(R,S)
00	$(1, -)$	01	$(1, -)$	(R,S)
00	$(-, -)$	02	$(-, -)$	(L,S)
01	$(0, -)$	00	$(0, -)$	(R,S)
01	$(1, -)$	01	$(1, -)$	(R,S)
01	$(-, -)$	03	$(-, -)$	(L,S)
02	$(0, -)$	09	$(0, 0)$	(S,S)
02	$(1, -)$	09	$(1, 0)$	(S,S)
02	$(-, -)$	09	$(-, -)$	(R,S)
03	$(0, -)$	09	$(0, 1)$	(S,S)
03	$(1, -)$	09	$(1, 1)$	(S,S)
03	$(-, -)$	09	$(-, -)$	(R,S)

(a) What does T do? Give as concise an explanation as you can. [8 marks]

(b) Convert T into a functionally-identical Turing machine that requires at least one less state than T. [8 marks]

(c) Explain, with examples as necessary, the following terms: *accept*, *decide*, and *recursively enumerable*. [9 marks]

10. (a) Define a language that is recursively enumerable but not context-free. [3 marks]

(b) Construct a deterministic Turing machine to accept your language from (a). [10 marks]

(c) Define a language that is recursively enumerable but not recursive. [6 marks]

(d) Give the outline for a program in any programming language that accepts the language from (c). [6 marks]

11. (a) Which of the following languages are recursive, which are recursively enumerable but not recursive, and which are not recursively enumerable? [14 marks]

i. the language of halting Turing machines
ii. the language of C programs that do not halt
iii. the language of C programs that do not halt when e is passed as input
iv. $L_1 = a^* = \{e, a, aa, aaa, aaaa, \ldots\}$
v. $L_2 = \{a, aa, aaa\}$
vi. $\overline{L_2}$, where $\overline{L_2}$ means the complement of L_2.
vii. $L_1 \cap \overline{L_2}$

(b) Can you enumerate the set of all possible languages over a finite alphabet? Prove your answer. [11 marks]

12. Prove the following theorems.

  (a) If a language is recursive, then it is recursively enumerable. [8 marks]

  (b) If L is a recursive language, the its complement $\overline{L}$ is also recursive. [8 marks]

  (c) The language of tours of length $\leq k \in \mathbb{N}$ of a weighted graph G is in $\mathcal{NP}$. [9 marks]

13. (a) Is the language of Turing machines that halt *recursively enumerable*? Prove your answer. [5 marks]

  (b) Is the language of Turing machines that halt *recursive*? Prove your answer. [10 marks]

  (c) Is the class of *recursively enumerable* languages closed under union and complement? Prove your answers. [10 marks]

14. Consider the *HALTS* problem of determining whether a given function *p()* terminates or not. Consider also the problem *CHANGE* to determine whether a given variable a changes value during the execution of function *q()*. Prove that *HALTS* and *CHANGE* are equally difficult problems. (Two reductions are required.) [25 marks]

15. What are the differences in terms of computability and computational complexity between the following machines.

  (a) A one-stack pushdown automaton and a two-stack pushdown automaton. [5 marks]

  (b) A one-stack pushdown automaton and a one-tape Turing machine whose tape is infinite in one direction only. [5 marks]

  (c) A two-stack pushdown automaton and a digital electronic Pentium (Intel Corp.) processor with 128 Mbytes of memory. [5 marks]

  (d) A one-tape Turing machine and a two-tape Turing machine. [5 marks]

  (e) A two-tape Turing machine and a three-tape Turing machine. [5 marks]

16. (a) Consider the following sextuple of language classes (regular, context-free, recursively enumerable, recursive, $\mathcal{P}$, $\mathcal{NP}$). Every language can be associated with a binary sextuple where symbol 1 denotes membership and 0 denotes nonmembership of the class in question. For example, if a language was in the first class and not in any of the others, it would be associated with the binary sextuple $(1, 0, 0, 0, 0, 0)$. State the binary sextuple associated with each of the following languages.

    i. The language of natural numbers divisible by 3. [3 marks]

    ii. The language of binary strings. [3 marks]

    iii. The language of Turing machines that go into a state 01. [3 marks]

    iv. The language of Java programs that execute a `print` statement. [3 marks]

    v. The language $L = \{w : w \in \{a, b\}^*, |w| \text{ is even}, w \text{ begins with an } a\}$. [3 marks]

    vi. The language $L = \{w : w \in \{a, b\}^*, |w| \text{ is odd}, w \text{ contains at least one } a\}$. [3 marks]

  (b) Is it true to say that finite languages are recursive and that infinite languages are not recursive? Prove that the finite language $\{aa, ab, bb, ba\}$ is recursive. Define a language that is not recursive. [7 marks]

17. (a) What does a reduction $A \leq B$ between two problems establish about their relative computability? What does it establish about their relative computational complexity? [6 marks]

(b) How could one use a reduction to prove nonmembership of a class? [4 marks]

(c) Give an example of a language that is recursively enumerable, an example of a language that is recursive, and an example of a language that is recursively enumerable but not recursive. [8 marks]

(d) Give an example of a deterministic finite automaton that accepts a regular language. Give an example of a nondeterministic finite automaton with fewer states that accepts the same language. [7 marks]

18. (a) What does the reduction $A \leq B$ establish about the relative computational complexity of two languages A and B? How can a reduction be used to prove nonmembership of a class? [5 marks]

(b) Prove that the class $\mathcal{P}$ is closed under complement. [10 marks]

(c) Prove that the problem SATISFIABILITY is in $\mathcal{NP}$. [10 marks]

19. (a) What is a polynomially-bounded (or polynomial-time) reduction? Illustrate with an example why these are of interest in computational complexity theory. [5 marks]

(b) The decision form of the travelling salesman problem asks: given a positive integer k, is there a tour of graph G of length at most k that visits each vertex exactly once? The optimisation form of the problem asks: what is the shortest tour of graph G that visits each vertex exactly once? Show how a solution to the former can be transformed into a solution to the latter, and calculate the approximate computational complexity of your transformation. [10 marks]

(c) Define co$\mathcal{NP}$ to be the class of languages defined as $\{\overline{L} : L \in \mathcal{NP}\}$. That is, co$\mathcal{NP}$ is the set of languages whose complement is in $\mathcal{NP}$. Prove that if $\mathcal{P} = \mathcal{NP}$ then $\mathcal{NP} = \text{co}\mathcal{NP}$. You are given that $\mathcal{P}$ is closed under complement. [10 marks]